

draft-ietf-taps-{arch,impl,interface} discussion

M. Welzl, on behalf of all document
editors: himself + Brian Trammell,
Tommy Pauly, Anna Brunstrom

<https://github.com/taps-api/drafts>

TAPS @ IETF 103

Connection & Message Properties

- #37: Appendix A: Experimental Transport Properties
 - "These are not part of the interface, and may be removed from the final document, but are presented here to support discussion within the TAPS working group as to whether they should be added to a future revision of the base specification."*
 - Direction of communication: uni-(s/r) / bidirectional
 - Suggest a timeout to the Remote Endpoint
 - Selection and protocol property (timeout value)
 - Traffic Category: query, control, stream, bulk
 - Size to be Sent or Received
 - Duration
 - Send or Receive Bit-Rate
 - Cost Preferences: no expense / optimize cost / balance cost / ignore cost

...which ones
do we keep?

Connection & Message Properties

- #208 and #218: re-structuring – "classification" disappears, it's implicit in the doc. Structure
 - Properties related to preconnection in pre-establishment section, etc.
- #243: The API suggest object, action, and event names. It should suggest property names as well.
 - Add "code" names for each of the Selection, (Generic) Connection, and Message properties in the respective subsections.
 - What should these names look like?
(see also registry discussion)

Cached State policies

#45 and #244: "Caching Context"

- Examples of cached state:
 - DNS query answers, TCP RTT metrics, TLS state
 - Different granularities: per-host, per-transport protocol, per-application
- As written now: a type of handle attached to the `NewPreConnection` call
 - Is this the right way to expose caching? Right granularity?
 - Some caching can happen in the transport system, without involving the application, some can happen inside the app

Extra issues (backup slide)

... if we have the time

Some other open issues

- #220: expose message dependencies in the API?
 - Have we finally agreed?
- #221: Connection.Receive
(minIncompleteLength, maxLength)
- #222: Reusing preconnections
 - It seems we have consensus on writing this;
who will?